Computer Science Department
CS632P – Topics Python Programming (CRN: 23765/23912)
Spring 2021

Project #1 (Due 17-Mar-2021)

In this assignment you will need to use Python's **argparse** and **logging** modules.

Device a Python app that will read the contents of any computer, and it should produce an output based on the input provided. The output will be written into a logging/logger log file, for a view on demand and verification. All the lines of the log file should be accompanied by time stamps, along with level message.

The 'main' **Python** script (list_content.py) should expect certain positional and non-positional arguments.
It should include a 'help' switch '**-h**' which will provide all possible switches and arguments.
It should also include a 'mutually exclusive group' for '**-v**' (verbose) or '**-q**' (quiet) mode.

The additional allowed switches/arguments to the script (list_content.py) should be:

1) '**-d**': To list all the drives of the machine with the following info:
   - Drive name / letter
   - Total number of directories
   - Total number of files
   - Total allocated, used, free storage

   A positional argument '**drv**' that will indicate the name of a single drive to report the above info.
2) **'l'**: To list all the folders in a given drive with the following info:
   - Folder name
   - Total number of files per folder
   - Total storage used per folder and sum of all storage for all folders.

   A positional argument '**fld**' that will pass the folder name to report the above info.
3) '**-f**': To list all the files of the machine with the following info:
   - File name
   - File type ('py', 'ipynb', 'exe', 'txt', 'csv', 'pdf', 'other')
   - File size
   - Date/Time stamp of the file

   A positional argument '**fil**' that will pass the file name to report the above info.
4) '**-t**': To list all the types of files exist in the machine with the following info:
   - File type
   - Total number of files per file type
   - Total storage used per file type

   A positional argument '**typ**' that will indicate the type of file to report the above info.

You should choose the various logging levels appropriately.
When listing info about folders and files the level should be **INFO**.
When the passing value for folder or file is not valid (folder or file doesn't exist) the level should be **WARNING**.
Any computation errors should be captured (e.g. total storage of files, folders, drives cannot be negative) and classified under the **ERROR** level.
When Python can NOT read the OS structure, the level should be **CRITICAL**.
Any other activity/message of the logger should have a level of **DEBUG**.